



Materia: Microprocesadores y microcontroladores

Maestro: García López Jesús Adán

Alumno: Christian Alfredo Montero Martínez

Practica 5b: Sección de Entrada/Salida

Fecha: 24/03/2017

Práctica No. 5b

Sección de Entrada/Salida

Objetivo: Acceso a los puertos genéricos del dispositivo programable PPI-8255 que se encuentra emulado en la tarjeta T-Juino.

Material:

- Programas TCC, TASM, TLINK y MKBINTJ
- Tarjeta T-Juino.
- Programa Terminal.
- Protoboard y compuertas lógicas (según diseño).

Equipo:

- Computadora Personal
- Programa emulador de terminal

Teoría

Interfaz de puertos paralelos 8255

Es un componente de bajo costo muy popular encontrado en muchas aplicaciones. Tiene 24 terminales para E/S, programables en grupos de 12, que se emplea en tres modos separados de operación. Este dispositivo puede interfazar cualquier dispositivo de E/S compatible TTL, al microprocesador 8088. El 8255 requiere una inserción de dos tiempos de espera (para un 8088 de 8MHz) cada vez que se activa por un comando de IN u OUT. Debido a que los dispositivos de E/S son inherentemente más lentos de cualquier forma los estados de espera utilizados durante las transferencias de E/S no afectan la velocidad del sistema.

La siguiente figura (Figura No.1) muestra el diagrama de las terminales del dispositivo:

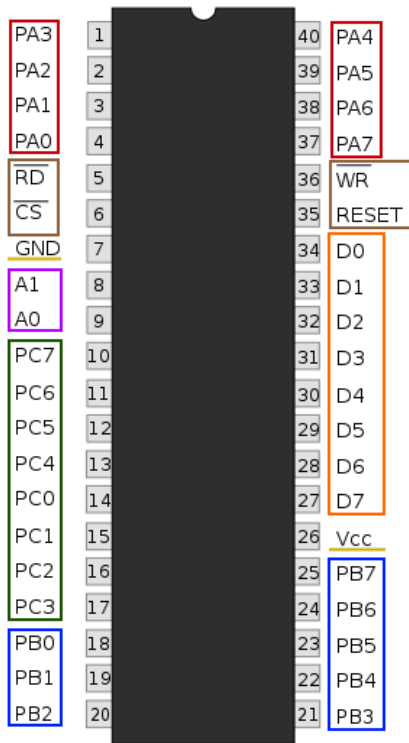


Figura No.1: Esquema de las terminales del dispositivo 8255.

En la Figura No.1 podemos observar los puertos de E/S indicados con diferentes colores, de color rojo se pueden observar las terminales correspondientes al puerto A, de color azul el puerto B y de color verde las terminales correspondientes al puerto C, mismas que se programan en grupos de 12 terminales. El 8255 se selecciona por su terminal CS' para programación y lectura o escritura a un puerto. La selección del registro es realizada a través de las terminales A1 y A0 que seleccionan un registro para operaciones de programación.

A1	A0	Función
0	0	Puerto A
0	1	Puerto B
1	0	Puerto C
1	1	Registro de comando

Tabla No.1: Asignaciones de puerto de E/S para el 8255.

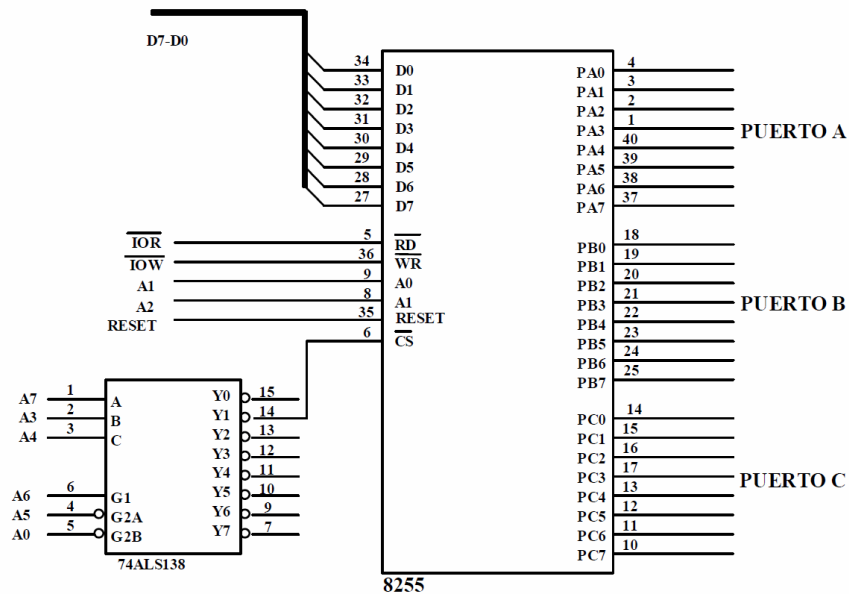


Figura No.2: Dispositivo 8255 interfazado al banco bajo del microprocesador 8088.

En la figura No.2 se muestra el dispositivo 8255 conectado al 8088, de tal forma que trabaja con puertos de E/S de direcciones de 8 bit. Nótese que todas las terminales del 8255 son conexiones directas al 8088 excepto la terminal CS'. La terminal CS' se decodifica y es seleccionada por un decodificador 74LS138.

Programación del 8255

Para la programación del dispositivo 8255 existen 2 bytes de control (A y B), el byte de control A programa la función del grupo A y B, mientras que el byte de control B pone (1) o reinicializa (0) los bits del puerto C (solo si el 8255 se programa en modo 1 y 2). Las terminales del grupo B pueden operar en modo 0 y 1. El modo 0 es el modo de operación básico de entrada y salida que permite que los terminales del grupo B sean programadas como simples entradas y conexiones de salida amarradas, mientras que el modo 1 es la operación de habilitación periódica para las conexiones del grupo B donde los datos se transfieren a través del puerto B y el puerto C proporciona las señales de reconocimiento.

Por su parte las terminales del grupo A pueden ser programadas como terminales de entrada y salida. La diferencia es que este grupo puede operar en los modos 0,1 y 2. La función del modo 2 es operar en modo bidireccional para el puerto A.

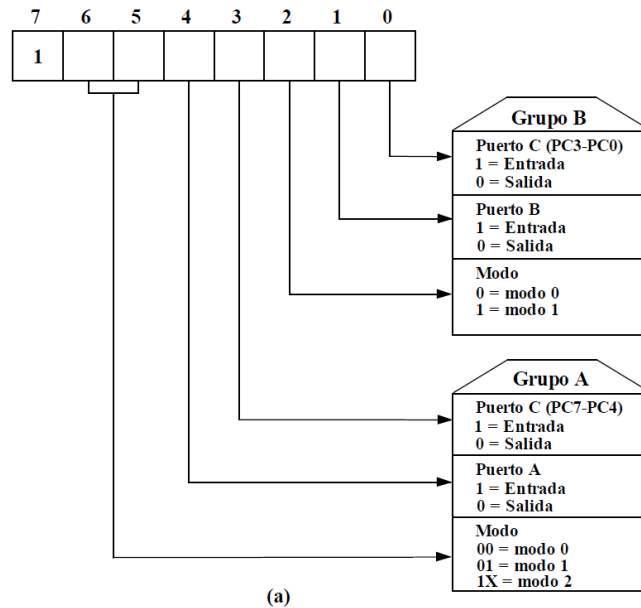


Figura No.3: Byte de control A para el registro de control del 8255.

Si se pone en 0 el bit 7 del byte de control, se selecciona el byte de control B. Este control permite que cualquier bit del puerto C sea puesto en 1 o 0.

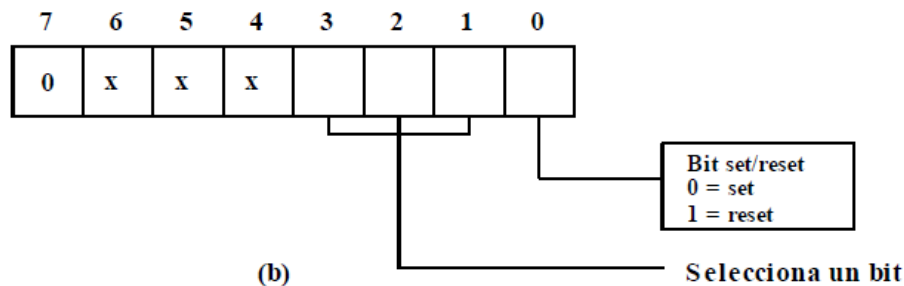


Figura No.4: Pone a 1 o 0 el bit indicado en el campo de bit seleccionado.

Modo 0: La operación del modo 0 ocasiona que el dispositivo funcione como una entrada reforzada o como una salida amarrada.

Modo 1: El modo de operación 1 ocasiona que el puerto A y/o B funcionen como dispositivos de entrada con amarradores, esto permite que los datos externos se almacenen en el puerto hasta que el microprocesador esté listo para almacenarlos. El puerto C se emplea en la operación del modo 1, no para los datos sino para las señales de control o reconocimiento que ayudan a operar uno o ambos puertos. El con entrada de habilitación periódica captura los datos de la terminal del puerto cuando se activa la señal de habilitación periódica (STB') justo en la transición de 0 a 1. La señal STB' ocasiona que se capturen los datos en el puerto y también activa las señales IBF (Reforzador de entrada lleno) e INTR (Requerimiento de interrupción). Una vez que el

procesador advierte que los datos están presentes en el puerto, a través del software IBF o del hardware INTR, ejecuta la una instrucción IN para leer el puerto (RD)'. Esta acción restaura los valores de IBF e INTR a sus estados inactivos hasta el próximo que el próximo dato es recibido en el puerto.

Modo 2: El modo 2 se permite solamente para el grupo A, el puerto A se convierte en bidireccional, permitiendo que los datos sean transmitidos y recibidos a través de los mismos 8 cables. Los datos acarreados bidireccionalmente son útiles al interfazar dos computadoras.

Desarrollo

Parte A: Programación y uso de puertos del PPI8255.

- 1) Crear archivo Prac5a.c que contenga a partir del listado 1
- 2) Realizar proceso para ejecutar el programa en T-Juino.
- 3) La siguiente función llamada SetBitPort manipula la información de un puerto dado para activar un determinado bit. Es decir mediante ella se puede activar (hacer uno) un bit del puerto. El número del bit está en el rango de 0 a 7 siendo el bit 7 es más significativo.
- 4) Diseñe las siguientes funciones para manipulación de bit de puertos.
Nota1: Es necesario implementar las funciones básica de E/S inportb() y outportb() en lenguaje ensamblador en archivo ASM independiente -- no usar funciones inline.
 - a) Función ClrBitPort la cual borrar un bit; es decir hace cero el bit de la posición num_bit del puerto dado por el parámetro Puerto.
 - b) Función NotBitPort la cual invierte el bit de la posición num_bit del puerto dado por el parámetro Puerto.
 - c) Función TstBitPort la cual retorna el estado del bit de la posición num_bit del puerto dado por el parámetro Puerto. Si el bit del puerto está en 0 lógico entonces la función retorna valor cero (0) de otra forma retorna valor uno (1).
 - d) Función ReverseBitsPort la cual retorna el estado del bit de la posición num_bit del puerto dado por el parámetro Puerto. Si el bit del puerto está en 0 lógico entonces la función retorna valor cero (0) de otra forma retorna valor uno (1).
- 5) Verifique el funcionamiento de las funciones del punto anterior realizando un programa de prueba.

Resultados

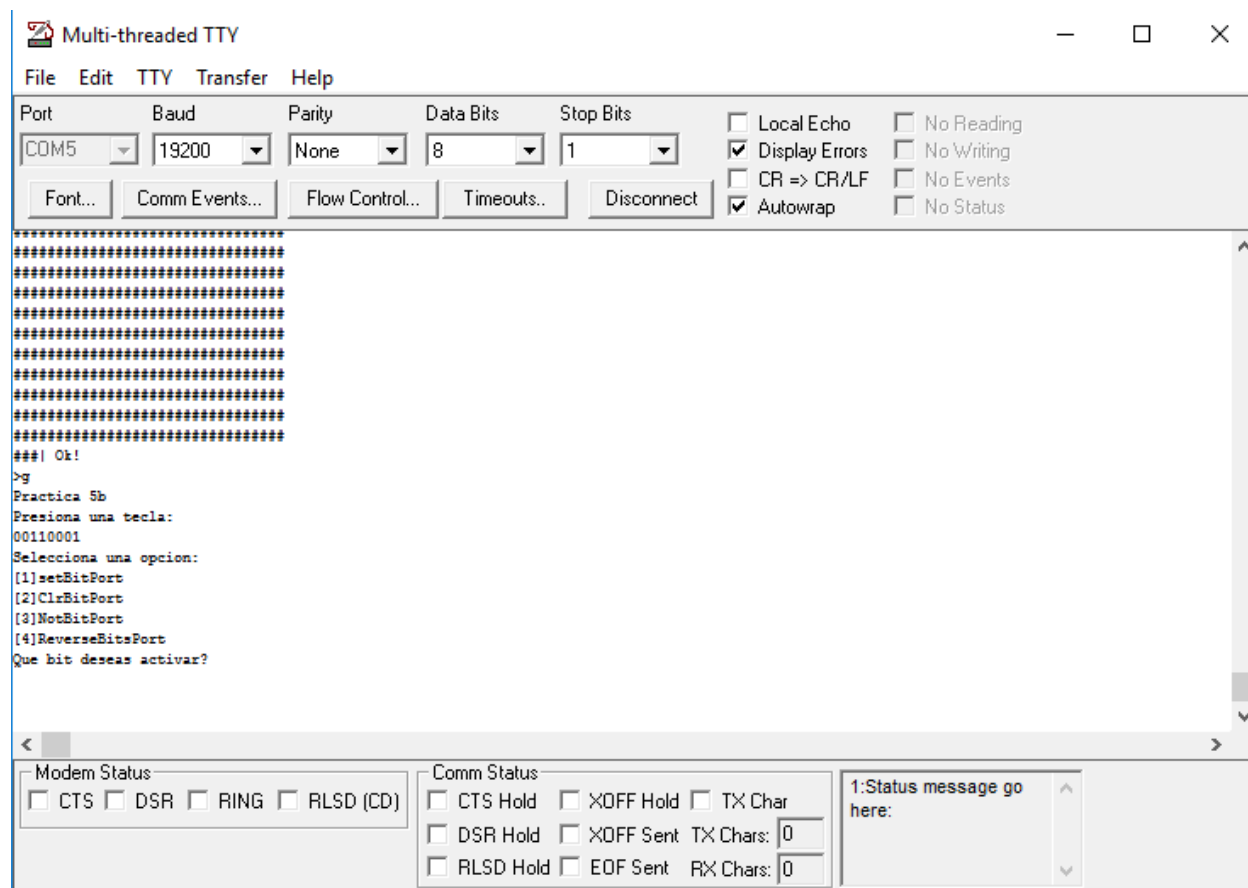


Figura No.5: Ejecución del software.

Como se puede apreciar en la Figura No.5 el programa de prueba de las funciones de manipulación de bits consiste en un menú con las 4 opciones, al seleccionar cualquiera de las primeras 3 el programa solicita al usuario que introduzca el número de bit que desea manipular, la opción 4 solo invierte la posición de los bits, el más significativo, por el menos significativo y así sucesivamente, el resultado de la manipulación de los bits, es enviado por el puerto B y se ve reflejado en los LEDS.

Conclusiones y Comentarios

El desarrollo de la práctica fue sencillo, las funciones que se programaron son básicas y no representan mucha dificultad, la única que tenía un poco más de complejidad era la de ReverseBitsPort sin embargo después de analizar un poco el problema encontré que se podía solucionar de manera muy sencilla utilizando las funciones que había desarrollado previamente en los otros incisos de la parte práctica (software) quedando una función sencilla de pocas líneas de código. En términos generales la práctica resultó muy interesante ya implica más interacción con los dispositivos de entrada y salida, la práctica pasada solo consistía en mandar una variable a un puerto mapeado en memoria, pero en esta se utilizan los puertos directamente, se manda una variable a los puertos para ver el resultado en los LEDS y después se manipula la información para hacer modificaciones en algunos bits específicos.

Bibliografía

Aguilar, L. Interfaz de periféricos programable 8255a-5 Universidad Autónoma de Baja California recuperado en Marzo del 2017 en: http://uabc-io.net/mdl/pluginfile.php/1293/mod_resource/content/1/interfaz_de_perifericos_programable_8255A-5.pdf

Brey, B (2009) *The Intel Microprocessors* Estados Unidos: Pearson Prentice Hall